IB 04 / 2815

Europäisches Patentamt | European Patent Office | Office européen des brevets

MAILED 13 OCT 2004
WIPO PCT

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

03292219.7

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

BEST AVAILABLE COPY

Anmeldung Nr:
Application no.: 03292219.7
Demande no:

Anmeldetag:
Date of filing: 09.09.03
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Schlumberger Systèmes
50, avenue Jean Jaurès
92120 Montrouge
FRANCE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Authentication in data communication

In Anspruch genommene Prioriät(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

H04L29/06

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL
PT RO SE SI SK TR LI

AUTHENTICATION IN DATA COMMUNICATION

Field of the Invention

This invention relates to authentication in data communication. In particular the invention relates to, but is not limited to, authenticating mobile stations and network servers communicating with each other through a network such as the Internet.

The example which will illustrate the invention is that of a mobile communication system comprising a mobile communication network and mobile stations. In this example, the network provides a service to a mobile station after authentication of the mobile station. The mobile station comprises a portable module such as a USIM card and comprises mobile equipment that is able to communicate with the network and that is able to communicate with the portable module.

Prior Art

The actual third generation (3G) standards (in particular TS 31.102 and TS 33.102) define the authentication protocol in a 3G network (known as AKA protocol, standing for Authentication Key Agreement) between the USIM and an Authentication Center (AuC).

In this framework, the card is sent a so-called authentication request made up of several data fields:

- a random challenge (RAND);
- a sequence number (SQN) or a concealed sequence number (SQN⊕Ak)
- a message authentication code (MAC);

AK being an anonymity key, the symbol ⊕ being the bitwise Exclusive or, MAC being a Message Authentication Code, SQN being a sequence number stating if the above request is a replayed request or not according to its value.

Upon receipt of these data fields, the card computes SQN (if required), checks the MAC and checks the freshness of SQN.

To compute the SQN(if required), the USIM computes:

- the anonymity key AK with a function f5 (RAND, K)
- Eventually, retrieves the sequence number SQN by way of (SQN ⊕ AK) ⊕ AK=SQN.

f5 is a key generating function used to compute AK

K is a Long-term secret key shared between the card USIM and the server.

Then, the card USIM also generates an expected message authentication code XMAC using the RAND, K,SQN ,an additional management field(AMF) and a authentication function f1.

Then,the USIM compares the XMAC with the MAC which was included in the authentication request If they are different, the client sends a user authentication reject message back to the handset with an indication of the cause and the client abandons the ongoing authentication procedure. In this case, the AuC may initiate a new identification and authentication procedure towards the client.

The USIM also verifies that the received sequence number SQN is in the correct range. The SQN may not differ more than by a predetermined amount of the SQN held by the USIM. If the USIM considers the sequence number not to be in the correct range, it sends synchronization failure message back to the AuC and abandons the ongoing procedure.

We will refer the above-identified standards for more details or more explanations on the above steps.

The code MAC (and therefore XMAC) is computed from the whole request data and the same authentication key. Its role is to ensure that the request data has

not been tempered during the transmission and also warrants the card that the requesting entity actually possesses the same authentication key as the card.

As the card is checking the integrity and authenticity of the data received from the server, the card computes said XMAC with a mechanism involving the data to check along with the authentication key K. Then, an attacker can force the utilization of the authentication key by sending to the card an authentication request with strategically chosen data. By various methods, such as side-channel and perturbation attacks, information is revealed, leading to the partial or total disclosure of the authentication key.

To be exploitable, most attacks require a given amount of authentication requests depending on the strength of the algorithm used to compute the XMAC. For each of these trials, the attacker must provide a dummy MAC (since it does not know the actual value of the key).

The Invention

The aim of the invention is to limit the number of consecutive attacks on a card USIM.

According to the invention, each time the two codes (MAC, XMAC) don't match, the authentication procedure is aborted, a counter keeping trace of the number of occurrences of abortion. In other words, the invention consists in associating the challenging step between MAC and XMAC with a failure counter in order to restrict the number of successive erroneous trials to a maximum amount, above which the key K is considered as compromised.

In this way, the number of malicious successive attacks is controlled. So, the card is a tamper resistant device, which is now more secure.

It will be easier to understand the invention on reading the description below, given as an example and referring to the attached drawings.

In the drawings:
Figure 1 represents an example of a data processing system S in which the invention may be applied.
Figure 2 is an example of an authentication failure counter management algorithm.

Detailed Description of Examples Illustrating the Invention

In order to simplify the description, the same elements illustrated in the drawings have the same references.

Figure 1 represents a system SYS including a user equipment communicating with a server SERV by way of a network NET such as Internet or private network. The user equipment consists of two parts: the Mobile Equipment ME and the Subscriber Identity Module CARD. The mobile equipment ME is the radio terminal used for radio communication between the user equipment and the server SERV. In our example, the card CARD is a USIM smart card that holds the subscriber identity, performs authentication algorithms, and stores authentication and encryption keys and some subscription information that is needed at the terminal.

The server SERV is able to provide a service to a mobile station after a successful authentication of the mobile station.

According to the invention, a counter controls the number of authentications aborted by the card. Preferably, the counter counts successive aborted authentications.

Figure 2 is an algorithm, which will illustrate clearly the invention. The authentication includes several steps (S1-S16)

In a first step (S1), the card receives an authentication request.

In a second step (S2), before checking the MAC, the card checks the erro counter:

- If the counter is zero (S3, S12), it considers that the key is compromisec and does not go further. In this case, the card returns a security erro message (step S14). After step S14, the authentication procedure is finished (S15)
- Else (S3,S4), it can use the key and verify (S5) the data provided MAC.
  - If the value expected by the card does not match with the one provided in the request, then the card decreases the error counter (S13) and sends a security error notification to the ME.
  - Else it checks the SQN of the request (S6), to ensure that it is not processing a request, which is being replayed.
    - If the SQN appears not to be fresh (S7,S10), then the card sends back a resynchronization token over the network (S10) as defined in the AKA. After S10, the procedure is finished (S11)
    - Else, if the SQN appears to be valid (S7,S8), then, in our example, the card resets the error counter to its maximal value(S8). After, the card can send a positive authentication result (S9). Step S16 is the end of the authentication procedure.

Once the error counter reaches zero, then the authentication key no longer can be used. Thus, it allows only a small amount of consecutive errors. The above-mentioned attacks require trials leading to MAC verification errors. Then the counter limits the number of trials and then the attack is biased.

For example, let's suppose that the maximal value of the counter is 3. Further, we assume that the initial value of the counter is 1. These 6 consecutive authentications illustrate several possible scenarios.

<u>1st Authentication: Counter >0 , Correct MAC , valid SQN</u>

Initial value of the counter: 1

- Reception of the authentication request (S1)
- As the counter is strictly positive (S3), a MAC verification is performed (S4).
- As the MAC is correct (S5), a SQN verification is performed (S6)
- As the SQN is valid (S7), the counter is reset to its maximal value, i.e. 3 (S8).
- The authentication result is returned (S9)

Final value of the counter: 3

<u>2nd Authentication: Counter >0 , incorrect MAC</u>

Initial value of the counter: 3

- Reception of the authentication request (S1)
- As the counter is strictly positive (S3), a MAC verification is performed (S4).
- As the MAC is incorrect (S5), the counter is decremented. The new value of the counter is 2 (S13)
- A security error is returned (S14)

Final value of the counter: 2

<u>3rd Authentication: Counter >0 , correct MAC , invalid SQN</u>

Initial value of the counter: 2

- Reception of the authentication request (S1)

- As the counter is strictly positive (S3), a MAC verification is performed (S4).
- As the MAC is correct (S5), a SQN verification is performed (S6)
- As the SQN is invalid (S7), a resynchronization token is sent. The counter is not modified. Its remains equal to 2.

Final value of the counter: 2

<u>4th Authentication: Counter >0 , incorrect MAC</u>

Initial value of the counter: 2

- Reception of the authentication request (S1)
- As the counter is strictly positive (S3), a MAC verification is performed (S4).
- As the MAC is incorrect (S5), the counter is decremented. The new value of the counter is 1 (S13)
- A security error is returned (S14)

Final value of the counter: 1

<u>5th Authentication: Counter >0 , incorrect MAC</u>

Initial value of the counter: 1

- Reception of the authentication request (S1)
- As the counter is strictly positive (S3), a MAC verification is performed (S4).
- As the MAC is incorrect (S5), the counter is decremented. The new value of the counter is 0 (S13)
- A security error is returned (S14)

Final value of the counter: 0

<u>6th Authentication: Counter =0 , incorrect MAC</u>

Initial value of the counter: 0

- Reception of the authentication request (S1)

- As the counter is equal to 0 (S3), the key is blocked (S12)
- A security error is returned (S14)

Final value of the counter: 0

The principal advantages of this solution are:

- The number of presentation of successive incorrect MAC is limited to the maximal value of the counter(Cf the above Authentications N°2,4,5,6).
- The total number of authentications is not limited as the counter can be reinitialized to its maximum value (Cf the above Authentication N°1).
- The replay of a correct authentication request does not reset the counter, as SQN is necessarily invalid and the counter unaltered (Cf the above Authentication N°3).
- Problems with the freshness of SQN do not risk blocking the card, as the counter is not decreased. (Cf the above Authentication N°3)

Several alternatives exist:

- The values of the counter are example.
- The counter management can differ. The counter can be increasing instead of decreasing, can evolve by step of any value, can be compared to any other value than 0 etc...
- The counter can count the total amount of authentication requests.
- The counter can count the number of incorrect MACs without possibilities to reset it to its maximum value.
- The counter can be reinitialized as soon as the MAC is correct (i.e. without any further checks such as the SQN validity).
- The counter can be decremented even if the MAC is correct and the SQN invalid.

CLAIMS:

1. Method of authentication in a system comprising two entities communicating between themselves by way of a network (NET), a first entity (CARD) authenticating a second one (SERVER) and data received from said second entity, each entity storing tho same secret key (K), said first entity receiving a message authenticating code (MAC) and other parameters (RAND, SQN, AMF, ...), the message authenticating code (MAC) calculation including the key K and said other parameters (RAND, SQN, AMF, ...), the authentication procedure consisting in challenging the received code (MAC) and an expected code (XMAC), the expected code being computed in using the received parameters and the key K stored in said first entity, characterized in that each time the two codes (MAC, XMAC) don't match, the authentication procedure is aborted, a failure counter storing the number of occurrences of abortion.
2. Method according to claim 1, characterized in that each time an authentication is launched, the smart card checks the failure counter before beginning the authentication procedure.
3. Method according to claim 1, characterized in that the card also performs a check on the (SQN) replay parameter, and if the authentication procedure is aborted due to an erroneous (SQN), the counter is not modified.
4. Method according to claim 1 or 3, characterized in that if the authentication procedure is aborted and the sequence number is fresh, the counter is reset to its initial value.
5. A smart card (CARD) able to authentify a remote entity (SERV) and data received from it, said card storing authentication algorithms, authentication and encryption keys, said card and said server storing the same secret key K, said card receiving a message authenticating code (MAC) and other parameters (RAND, SQN, AMF, ...) from said server, calculation of (MAC) including the key K and said other parameters, the authentication

procedure consisting in challenging the received code (MAC) and an expected code (XMAC), the expected code being computed in using the received parameters and the key K stored in the card, characterized in that said smart card stores a failure counter able to store the number of occurrences of abortion of authentication procedure.

Abstract

Method of authentication in a system comprising two entities communicating between themselves by way of a network, a first entity (CARD) authenticating a second one (SERVER) and data received from said second entity, each entity storing the same secret key K, said first entity receiving a message authenticating code (MAC) and other parameters (RAND, SQN, AMF, ...), the message authenticating code calculation including the key K and said other parameters (RAND, SQN, AMF, ...), the authentication procedure consisting in challenging the received code (MAC) and an expected code (XMAC), the expected code being computed in using the received parameters and the key K stored in said first entity, characterized in that each time the two codes (MAC, XMAC) don't match, the authentication procedure is aborted, a failure counter storing the number of occurrences of abortion.

Figure 2.

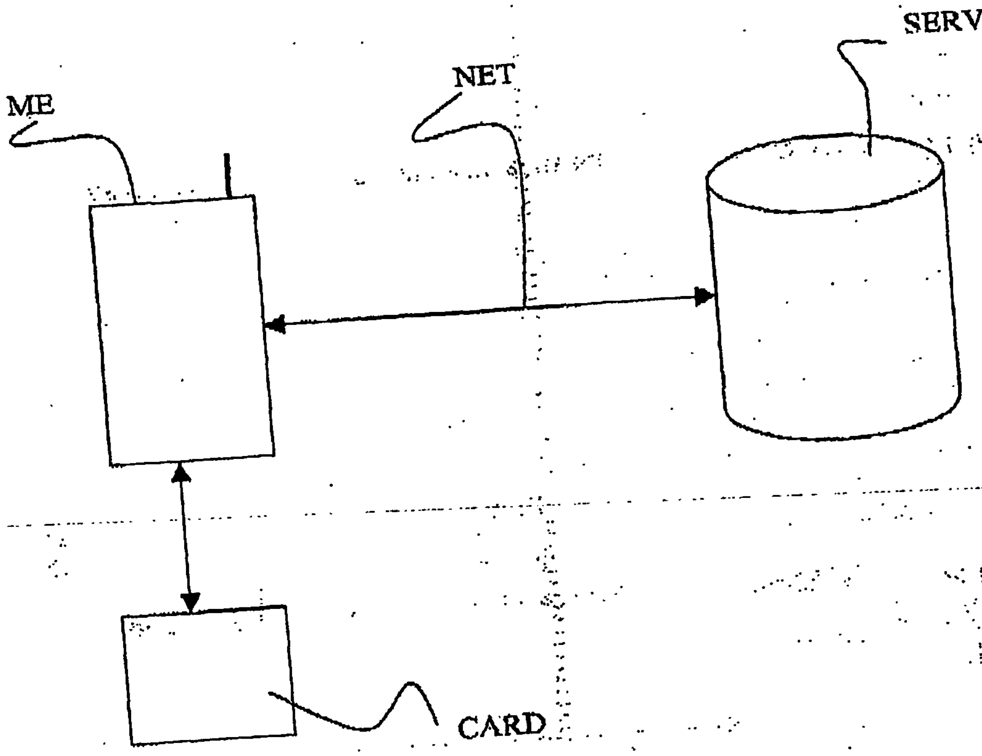

Figure 1

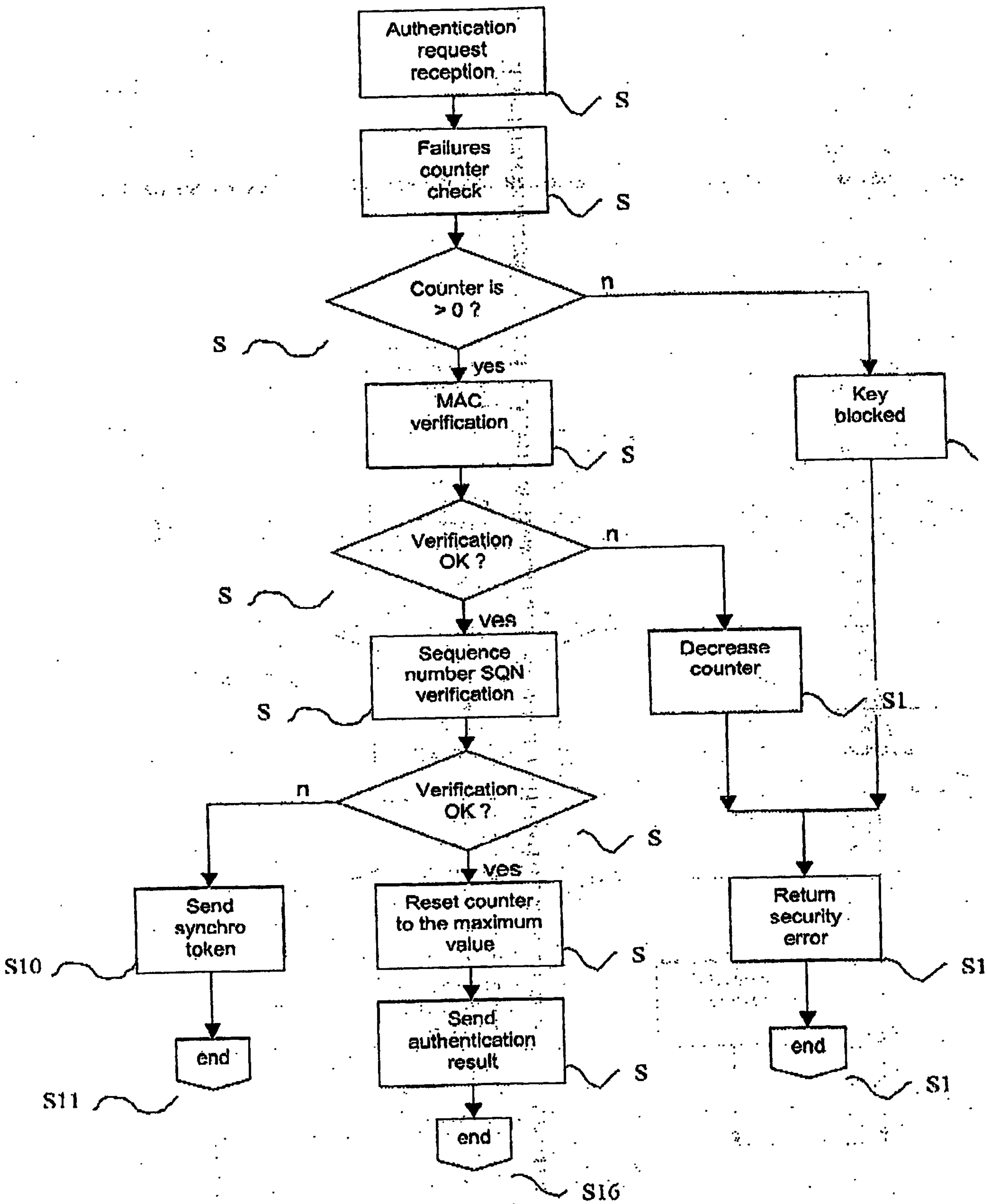
Authentication request reception
S
Failures counter check
S
Counter is > 0 ?
n
yes
S
Key blocked
MAC verification
S
Verification OK ?
n
yes
S
Sequence number SQN verification
S
Decrease counter
S1
Verification OK ?
n
yes
S
Send synchro token
S10
end
S11
Reset counter to the maximum value
S
Send authentication result
S
end
S16
Return security error
S1
end
S1

Figure 2

PCT/IB2004/002815